Technical Report ITL-94-6
August 1994

US Army Corps
of Engineers
Waterways Experiment
Station

Development of Executive Information Systems

by Ernest H. Boswell, Jr.

WES

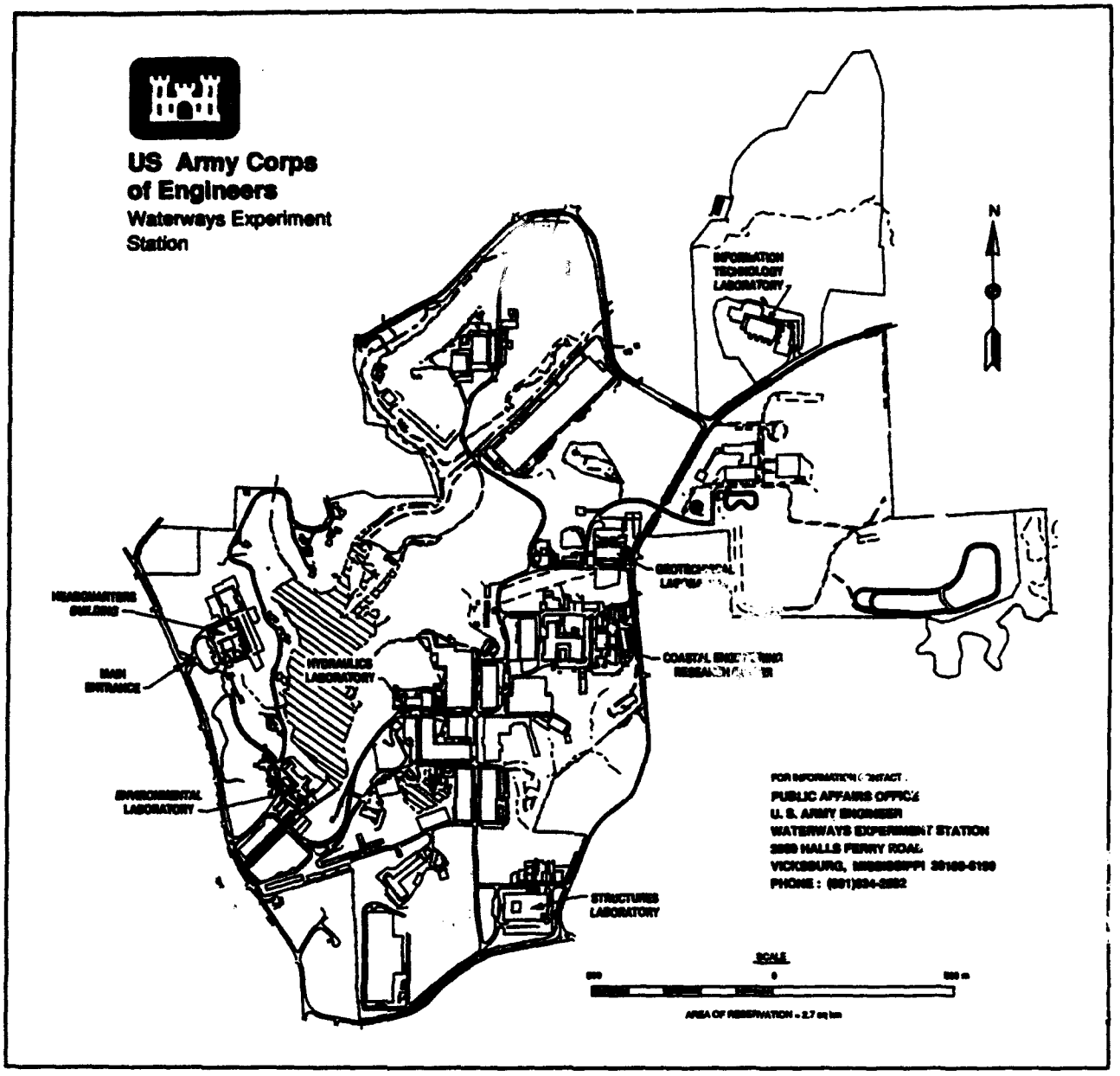
Technical Report ITL-94-6
August 1994

Development of Executive Information Systems

by Ernest H. Boswell, Jr.

U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final report



Waterways Experiment Station Cataloging-in-Publication Data

Boswell, Ernest H.
Development of executive information systems / by Ernest H. Boswell, Jr.
40 p. : ill. ; 28 cm. -- (Technical report ; ITL-94-6)
Includes bibliographic references.
1. Management information systems -- Evaluation. 2. EIS (Information retrieval system) 3. Relational data bases. 4. Management -- Data processing. I. United States. Army. Corps of Engineers. II. U.S. Army Engineer Waterways Experiment Station. III. Information Technology Laboratory (US Army Corps of Engineers, Waterways Experiment Station) V. Title. VI. Series: Technical report (U.S. Army Engineer Waterways Experiment Station) ; ITL-94-6.
TA7 W34 no.ITL-94-6

Contents

Preface

This technical report documents a review of Microsoft Windows-compatible software products for use in developing Executive Information Systems. It is not intended that this report be considered a comprehensive examination of all available software that could conceivably meet this need. The software included in this report was selected because it was either being used for this purpose on current projects, commonly available, or representative of new capabilities. The sample application was devised before the selection of software to prevent bias either for or against any single package.

The research and development was done and this report written at the U.S. Army Engineer Waterways Experiment Station, Information Technology Laboratory (ITL), Computer Science Divison, Systems Modernization Unit, by Mr. Ernest H. Boswell, Jr. Funds were provided by the Corporate Database project.

This work was performed under the supervision of Ms. Barbara Comes, Chief, Systems Modernization Unit, Dr. Windell Ingram, Chief, Computer Science Division, and Dr. N. Radhakrishnan, Director, ITL.

At the time of publication of this report, Director of the Waterways Experiment Station was Dr. Robert W. Whalin. Commander was COL Bruce K. Howard, EN.

1 What is an Executive Information System?

The essential purpose of an Executive Information System (EIS) is to provide management a meaningful and timely summarization of pertinent information, presented in a useful manner, with a means of navigation to increasing levels of detail. It should also be intuitive, making written documentation or substantial training unnecessary. Targeting management, particularly senior management, as the prime user of an automated information system entails several potential pitfalls. There may be considerable resistance, due to a lack of familiarity with computers, vague understanding of the potential benefits of the system, unwillingness to change the status quo, or negative experiences with previous information management projects. There are also possible internal political ramifications, if the development effort fails to deliver as promised.

Data management is a critical issue. The value of any information system depends on the quality of the data. This is particularly true for an executive information system, due both to the intended users of the system, and the need to gather data from multiple hardware and software platforms. An EIS may highlight existing problems, with a corresponding loss of confidence in both the EIS and the underlying data. Since data must often be gathered from various divisions within the organization, the owners of the source data must be included in the quality assurance process. There are two reasons for this. The source data must be updated on a schedule that coincides with the requirements of the EIS, and integrating data originally intended for different purposes may require close interaction between the data administrators of the divisional databases, at least during the development phase.

An executive information system has several characteristics. These include, but are not necessarily limited to:

a. transparent access to many sources of data

b. drill-down/drill-up functions

c. a user interface straightforward enough for casual users to understand

d. the capability to provide meaningful information to users at any level of the target organization

e. presentation of key performance measurements

f. clear system navigation capabilities

Probably the biggest obstacle to developing an effective EIS is the definition of the scope. Should the user be able to update the source data? Is there a need for "what-if" analyses? Should electronic mail or messaging be integrated? Is an ad-hoc query capability needed? Because the judgement of the success or failure of an EIS tends to be very subjective (am I receiving meaningful information from this?), requirements definition can be more difficult than for other systems development activities. Since requirements definition frequently becomes an iterative process, it is essential that "expert users" be deeply involved. This factor gives added weight to the selection of the software used as the platform for the EIS. The "look and feel" of the software should be as familiar as possible for these users. Microsoft Windows provides an excellent environment, since the Windows-based word-processing and spreadsheet programs already in use by many of these "expert users" have made them familiar with the techniques of Windows operations. For that reason, all of the software programs evaluated in this review are Windows-based.

As Microsoft Windows has become a standard desktop computing environment, the graphical capabilities of Windows and the power of 80386- and 80486-based personal computers have combined with network technology to provide powerful EIS functionality. The client-server model and software that takes advantage of client-server techniques have become a standard for EIS development. With the advent of spreadsheets and database management systems written specifically for the Windows environment, the dividing line between EIS software and more traditional software has become increasingly blurred. In many cases, however, a distinction can be made between software that provides certain functions by default, and software that can be made to provide those functions. Several representative tools for EIS development were evaluated and a prototype system developed with each using a standard set of requirements. Examples of screens for each sample system can be found in Appendix B.

2 Software Packages Evaluated

Six software packages were selected. Two are products developed specifically for use in developing EIS-type applications: Lightship, from Pilot Software, and Forests And Trees, from Trinzic Corporation. Two packages are spreadsheets: Quattro Pro for Windows, from Borland International, and Excel, from Microsoft. Also evaluated were a Windows database management system, Access, from Microsoft Corporation, and Oracle Card, from Oracle Corporation, a graphical database application development tool.

Lightship
(Lightship Version 3.2.0 and Lightship Lens Version 3.2,Pilot Software, 40 Broad Street, Boston, Massachusetts 02109)

Forests And Trees
(Forests And Trees Version 3.0, Trinzic Corporation, *101* University Avenue, Palo Alto, California 94301)

Excel
(Excel Version 4.0, Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399)

Quattro Pro
(Quattro Pro for Windows Version 1.0, Borland International, Inc., 1800 Green Hills Road, Scotts Valley, California 95067-0001)

Access
(Access Version 1.1, Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399)

Oracle Card
(Oracle Card Version 1.1, Oracle Corporation, 500 Oracle Parkway, Redwood Shores, California 94065)

3 Description of Review Process and Sample Applications

The source data for all sample applications is the Civil Works Life Cycle Project Management Reporting System (LRS) database. This database is housed on a Control Data Corporation 4680. The host database management system is Oracle 6.0.33.3.2. For those reviewed packages that could not access Oracle databases, a set of dBase files was created. The LRS database provides the capability to track project costs, schedule, and issues on a monthly basis. Each project can have multiple milestones and cost features for each month. The database contains separate tables for project header data, schedules, cost features, and project status. A partial data dictionary, for tables used in this evaluation, is included in Appendix D.

The sample applications were chosen to demonstrate each package's ability to provide sums, averages, and counts for data groupings, and to present charts of the results. Not all applications were developed for each package. Since the objective of this evaluation was to determine the suitability of a given package for use in developing EIS applications, if it was determined that a particular package was not appropriate for a given application, the application was dropped for that package. The sample applications are:

a. Average Overhead, calculated by district and division;

b. Number Of Reported Milestones Per Project, averaged for district and division;

c. Total Cost Of Active Projects, summed for district and division.

4 Evaluation Criteria

The following criteria were used in evaluating the software:

a. Database compatibilities

b. Difficulty and flexibility of methods of accessing data in external databases

c. Screen development capability

d. Graphing capabilities

e. Ease of development

f. Ease of maintenance of completed application

g. Distribution requirements

Each package was given a numeric score for each item. Scores ranged from 1 to 5, with 1 being the lowest (least functional) and 5 being the highest.

5 Summaries of Developed Applications

Lightship

Lightship is designed to facilitate the development of Executive Information Systems and other types of graphical, query-only database front-ends. It can access and display data from many external data sources, although it cannot use other Lightship files as a data source. The basic units of a Lightship application are screens, objects, data sources, actions, and variables. The screen is the foundation of the application. Each screen is a separate file, and you link screens with navigation commands to create an application. Because of this, modification of an individual screen can occur without disturbing other modules in your application.

Lightship is a powerful, flexible tool that is also somewhat difficult to use, especially when compared to the other packages in this evaluation. More than any of the other tools, development projects using Lightship are programming-intensive tasks.

The screen is built by creating objects. An object is classified as document, chart, format, image, hotspot, menu, text, or draw. These objects have the following characteristics:

- Document objects display data from external sources or data that the user enters into the document
- Chart objects are dynamically linked to document objects, and display document data graphically
- Format objects format and align document object data
- Image objects display an image from an external source (.BMP, .PCX, clipboard paste, or DDE)
- Hotspot objects perform actions, and can be invisible or displayed as a button

- Menu objects perform actions from the Windows menu bar
- Text objects display simple text, for screen headers or other information boxes
- Draw objects draw simple objects such as lines or borders to enhance the screen display

Document objects are the centerpiece of most Lightship screens. In Lightship parlance, a document is any data, whether it is text from an ASCII file or records from an SQL database. The document data is formatted by creating a format object and then applying the format object to all or part of the document object. Screen layout is accomplished by creating image objects, text objects, and draw objects.

Lightship operates on the basis of levels. There are three possible levels when running Lightship - Author, which is used when developing applications; Browse, which is used when running the application; and Capture, which is similar to Browse, except that the user does not have access to Windows. In Capture mode, the user cannot change to another level and exits to DOS when the application is terminated. One quirk of Lightship is that the menu and hotspot objects only function at the Browse and Capture levels, meaning that when you create one of these objects, you must change to the Browse level to test it.

Lightship provides access to ASCII text files, Windows DDE servers, user-entered data, external sources through user-defined DLLs, or external databases via Lightship Lens, an allied software product that allows SQL queries against external data sources. You select the fields you want and set selection criteria. If your data contains a numeric field, Lens assumes you will be consolidating the data in some way, and asks you to specify how you will do this (sum, average, count, minimum, maximum) for each numeric field. There is no way to turn off consolidation, so if you just want to query your data, consolidation becomes an unneeded but required overhead. Lens allows you to specify which results will be displayed, so you can eliminate the consolidation result. When you have run your query, you can save the results in a file that Lightship calls a cache. Lens also allows you to access your data by directly entering SQL statements. This method does not perform any consolidation - the results of your query are the contents of the document window. When using SQL, however, you cannot save the data into a cache, so the query is re-executed each time you access that document object. If you are querying a large database, this can cause a significant delay in presenting the screen. The queries supporting the demonstration applications were mainly run using the SQL interface, which caused navigation between screens to be quite slow, with delays of 3 - 5 minutes sometimes occurring.

Once the data was retrieved, formatting was simple, although the document display had a "quirk." If the rightmost field was a numeric field with several

digits to the right of the decimal, and was re-formatted to a fixed length of 2, the remaining digits would sometimes show up several spaces to the right of the field. I could not remove these digits - the only remedy was to shrink the document window so that those digits were outside the window area. The format object is created and applied to the column or columns by simply marking the affected column(s) with the mouse. If the document window is subsequently resized to display more rows, however, you must return to your format object and re-mark the target of the format.

Lightship's charting tools were generally very good. There was one particularly effective feature that no other package offered - scrollable bar charts. The bar chart displayed only the rows displayed in the corresponding document object window, so that as you scrolled up or down through the document window, the chart automatically scrolled through the display. If you have multiple charts associated with a document object, each chart scrolls accordingly. This is a unique answer to the problem of illegible bar charts due to too many data items in the series.

Navigation between screens was fairly simple to implement. There are several ways to do this - buttons, "hot spots", or menus created on the Windows menu bar. In each case, you can perform a variety of actions based on selection of the button, "hot spot", or menu item. Lightship uses the concept of a retrace path, which maintains the name of each open screen in the order in which it was opened. You can return to the screen that called the current screen, select any screen currently in the retrace path, open a new screen and add it to the retrace path, or clear the path.

Each user must have a copy of Lightship to run the distributed application. If your EIS will have many potential users, this could require a heavy financial commitment to a product that has no other uses. Despite this, Lightship could be a very effective tool in the right environment. If the need is for many different screens, each running directly against the source data, and the necessary data aggregation matches the consolidation methods that Lens performs, Lightship may be the best software in this group. In general, however, other tools can provide the same or better capabilities with less effort and less expense.

Forests And Trees

Forests And Trees is a data access and reporting tool specifically designed to produce EIS-type applications. Where Lightship stores each screen in a separate file, Forests And Trees holds all parts of the application (queries, charts, data, etc.) in one file called a "view file." This makes it easier to distribute applications, since you only have one file to move, but it can be a hindrance when you only need to change and distribute one part of the application.

The view is the basic building block of Forests And Trees. View type is determined by the result of the underlying calculation:

- Standard views display a table containing the data returned by a query, formula, or DDE request, or a graph based on that data
- Cross-tabulation views are standard views which have been re-oriented to show the data with new totals or a different perspective
- Edit field views display or accept user-input text
- Picture views display images contained in a bitmap (.BMP) or icon (.ICO) file, and can function as a button when the mouse is clicked on the view
- Button views display a button that performs an action when clicked
- Hot spot views are transparent views that perform an action when clicked. They are generally used as overlays.
- List views display a list of items that can be selected by the user.

You begin creating an application by creating a view. The source for a view is called the provider. A provider can be a view within the active view file or another Forests And Trees view file, or an external data source. Once built, the views can be combined into groups. Groups can be opened and closed by clicking on the group icon, and view windows can be maximized or resized using the standard Windows methods. You can also specify a bitmap as the backdrop for the opened group. A key feature of Forests And Trees is the drill-down button, which is automatically generated anytime you create a view which uses another view in that view file as the provider. You can then navigate up and down through your linked views using the drill-down button. An information tree is also automatically created, which allows you to display and sccess, in organization chart style, the underlying views.

Accessing data in other databases or in other views in the current (or other) view file, and building the query to return the desired information, is very easy. The Edit Query window (Figure 4) displays all available files in the provider (data source) that was specified when the view was defined. If you click with the mouse on the box to the left of the file name, all fields in that file are displayed. If you want, you can browse the file (or any file in that provider). You can enter the SQL statement directly (all data access in Forests And Trees is via SQL), or you can build the statement by using the mouse to click on the SQL language elements displayed in the window. You can test your query without leaving the Edit Query window, retrieving only enough rows to fill the window - for queries against large tables, this can save considerable time. When you have completed the query, you can even have Forests And Trees automatically format the statement, placing each element on

a separate line, capitalizing, and so forth. When you are finished, you exit the Edit Query window and calculate the result. When you recalculate a query, all queries built from that query can be automatically recalculated. You can also schedule recalculations.

Since Forests And Trees holds all objects in one view file, on a large application you can easily have dozens of views in the view file. Fortunately, you can choose to hide views so they aren't displayed. You can then display and hide views using triggers attached to buttons, views, or view columns. Triggers are a powerful object within Forests And Trees. A trigger, executed by clicking the mouse or setting an alarm based on a value or range of values, can open views or groups, calculate a view or group, display messages, launch another Windows application, open another view file, or execute formulas. If the trigger is set for a column, the value clicked can be placed into a substitution string for use in calculating a view. Triggers can be set for a view, a column, or an alarm. This allows one view to spawn a variety of processes based on actions by the user. In the demonstration application, triggers were used to provide navigation between screens by assigning them to buttons and picture views.

Charting is as good as any package evaluated. You can cycle charts in your view window, which lets you switch between data and chart by clicking on a button. Also, if you have widely scattered values, you can choose to limit the range of your chart, then switch between the full chart and the limited chart by clicking the Automatic Range Selection box in the graph definition window.

Overall, Forests And Trees is a powerful product with few weaknesses and many strengths. One problem with Forests And Trees lies in maintaining the view file, since this can become quite large for complex applications. However, this can be mitigated somewhat by designing the application to use several smaller view files. For many EIS applications, this may the best tool of this group. However, if you need to update any of the source data, Forests And Trees is the wrong tool, since it is incapable of writing data to the original database. Another disadvantage for Forests And Trees is the need to buy a separate copy of the software for each user. There is no distribution kit available, and while pricing for subsequent copies after the first license is discounted, it is still quite high for a single-purpose package.

Excel

Microsoft Excel is a spreadsheet, a type of software that simulates a paper worksheet with a matrix of rows and columns. While you are constrained by the inherent limitations of spreadsheets in general, Excel provides functions that allow the developer to disguise the spreadsheet look and feel in many cases. Access to external databases is provided by a special version of the

Q+E database editor (in version 5.0, this will be replaced by Microsoft's Open Database Connectivity driver). Although not as elegant as Lightship or Forests And Trees, it will accomplish the task, allowing you to set selection criteria, perform joins, and build SQL statements. Data is retrieved into a block of cells in the spreadsheet, where it can be manipulated like any other spreadsheet data. This allows "what-if" activities to be performed on the data, something Lightship and Forests And Trees cannot provide. However, you cannot update the source file with the results of the "what-if" analysis.

The application developed for this evaluation consisted of only one spreadsheet, with the screens using separate blocks within the spreadsheet file. This created some problems when resizing rows and columns to make bitmap images fit the screen, because it would sometimes throw other blocks off. Multiple spreadsheet files, combined into what Excel terms a workbook, could have been used to resolve that problem. That approach, however, would have required maintaining several files. Managing the data within the spreadsheet was more difficult than with Lightship, Forests And Trees, or Access. Because Excel is a spreadsheet, you are forced to use blocks to hold the data, whether for display or further manipulation. This tends to scatter your data around the spreadsheet, and while you can assign labels to blocks to simplify the process, data management can be tedious for large files.

Although comparatively limited, formatting was easy. Excel allows you to hide the grid lines, so you can give the screen the appearance of having a display window within the main display. If you have more than one block of data on the screen, however, you can't scroll up or down through them independently. You also have to be careful when resizing rows or columns, to avoid an accidental change to another display area that shares the same row or column. Navigation between screens requires creating buttons and writing macros that you attach to the buttons. It is not as simple a process as in the two EIS-specific packages, and maintaining the macros can be confusing unless you use a consistent and clear naming convention when creating the macros. Excel stores macros on a separate worksheet, and it is easy to accidentally create multiple macro worksheets. In some cases this may actually be desirable, for grouping of logically related macros, but it can also lead you to modify the wrong macro.

Charting is very good, generally on a par with Lightship and Forests And Trees. Excel uses Microsoft's Chart program, as does Microsoft Access, so there is the potential for developing a sophisticated application using both programs and having charts with the same look and feel. You have more control over certain chart features and labels than with any of the other packages, and the charting software itself was easier to use. And while the Chart program is actually external, the integration is seamless.

Incorporating graphics for use as backdrops for your screens is possible, although the result is not as clean as with the non-spreadsheet packages. The simplest way to do this is to paste an image, using the Windows cut-and-paste

functions, into a block of cells. This creates a graphic object, which can be resized or moved as needed. The object continues to reside in the block of cells, however, so as rows and columns are resized, the size of your image will be resized. While the process is manageable, you can end up with unintended white space at the side or bottom of the screen. Another possible drawback occurs when you are scrolling through a list of data. With a spreadsheet, you are actually moving down or across through the cells. If you want the headers to stay positioned above the data, you must lock the header rows in position; this can affect other parts of your spreadsheet file. Also, you may have rows of different heights, if you used a larger or smaller font in another part of your application that shares the same row.

Experienced spreadsheet developers should have no problem creating an application that goes far beyond the normal expectations of a spreadsheet. It does, however, require considerable effort, and the overall effect is not as polished as with the non-spreadsheet software. While Excel does not have a distribution kit, spreadsheets have become one of the most popular software products today, and many organizations use Excel. Because of this, the lack of a distribution kit should not be a major disadvantage.

Quattro Pro

Like Excel, Quattro Pro is a spreadsheet. They are very similar in many respects. One major difference is the way Quattro handles multiple spreadsheets. While Excel has a workbook concept, combining separate spreadsheet files, Quattro uses the metaphor of a notebook, with spreadsheets being represented by pages. Unlike Excel, Quattro stores all pages in one file.

A utility called Database Desktop provides access to external data. Quattro can also directly import data from Lotus 1-2-3, Microsoft Excel, and ASCII text files. You use Database Desktop to build and store queries, set selection criteria, perform joins between tables, and create permanent files called Answer Sets. Results are stored in a database file (either Paradox or dBASE format) which Database Desktop calls an answer table. You can also edit data in the source file. However, you are limited to accessing Paradox and dBASE files. Database Desktop is not as easy to use as the Q+E database editor included with Excel. It was the least useful data access tool.

Once the data was retrieved, formatting was simple, although, as with Excel, somewhat limited. Disguising the spreadsheet look and feel is difficult. Other than removing grid lines and scroll bars, there wasn't much way to do this. This is particularly true when scrolling through lists of data. Quattro does have a nice feature that does adapt itself well to EIS applications, however. Each worksheet is handled as a page in a notebook, and a tab for each worksheet is displayed along the bottom of the display. The tab names default to A, B, etc., but you can rename them very easily and provide

meaningful names for each page. You can switch between worksheets by clicking on the tab for that page. You can also create buttons and attach macros to them as a navigation method. Incorporating graphic images for backdrops requires the same process as with Microsoft Excel, and has the same limitations.

Charting of data was generally good, although not as good as the other packages. In particular, display of labels was inferior. There was apparently no way to align labels vertically, and if labels were bunched too closely, some were dropped. You could stack labels in a cascading 2- or 3-row manner, but the result was often crowded and difficult to read.

Overall, while Quattro Pro is a very powerful spreadsheet, it's data access limitations and comparative deficiencies in charting make it a less attractive choice for this type of development project. As with Excel, you are taking software intended for another purpose and trying to make it conform to your needs. A few years ago, this would have been impossible with spreadsheets. Now, however, it is possible to produce a reasonable application that gives little indication that a spreadsheet lies beneath. While you probably wouldn't use Quattro Pro or Excel to build a full-fledged executive information system, either could have a place as the basis for a small EIS or as a part of a larger system. There is no distribution kit; however, this would not be a major problem if you already use Quattro as your standard spreadsheet.

Access

Access is a database management system developed specifically for the Windows environment. It can be used to develop complete, stand-alone database applications, or graphical front-ends to existing databases. Access uses the standard relational database concept of a table to store data. When queries are run against tables, whether internal Access tables or external tables, the result is placed in a dynaset. A formatted screen for displaying data is a form. There is also a report generator, for producing printed reports. Like Forests And Trees, Access stores all data, forms, reports, queries, and graphs in one file. You can create buttons and attach macros to them to perform various functions, including navigation between forms. This allows the creation of sophisticated, but easy-to-use applications.

Access can utilize data from external sources in two ways. You can import data from Excel, Lotus, Paradox, dBASE, Foxpro, Btrieve, or SQL Server and store the data in an Access table. You can also attach to external tables and use the data directly in an Access form or report. When attaching to an external source, you can add records, modify records, or delete records in the attached file. In addition to importing and attaching files, you can build a query against an Access table or external file or table. The query window uses Query-By-Example (QBE) to build the SQL statement. You can select the

fields with the mouse and drag the field into the QBE grid, where you can set selection criteria and sort properties. If you need to join tables, you connect join fields by clicking the first field and dragging the mouse to the join field in the second table. You perform aggregation operations by clicking the Totals button on the Toolbar, which allows you to select an aggregation operation from a pick list. If desired, you can edit the SQL statement by clicking on View and selecting the SQL option. The query results can then be saved in an Access object called a dynaset.

Because Access is a full development environment, there were more options for data display than in any other package. Building forms is easy, and made easier using Microsoft's Wizard technology. Wizards are context-sensitive help windows that steer you through the building process for forms, reports, charts, or any other Access object. You can let the Wizard build the basic form or report, then enter Design mode to customize it, or you can bypass Wizards and design the form from scratch. You can create either single-record or tabular forms, specify the appearance, create sub-forms, and imbed graphic objects.

Like Excel, Access uses Microsoft Chart as its charting tool. Surprisingly, the interface to Chart from Access is not the same interface you see with Excel. There are a few chart choices missing, although there are still plenty of possibilities. Otherwise, your formatting options are the same. As with Excel, integration with Chart is seamless.

Navigation was performed by creating buttons and attaching macros to them. This could also have been done by creating custom menus on the Windows menu bar and placing menu commands on them, also by attaching macros. In addition, Access includes a programming language called Access Basic, which can be used to create modules to handle processing too complex for macros.

Access can match any of the reviewed packages almost on a feature-for-feature basis. Where Access doesn't automatically provide a given capability, you can easily provide it with a macro. Combined with the programming tools available, you have the potential for an application that goes far beyond what you could build with any other reviewed product. Access is the only package that has a distribution kit available. Distribution of an Access-based EIS would, therefore, be less expensive than applications developed using any of the other reviewed software. In many ways, it has no real competitor in this group of software.

Oracle Card

Oracle Card is an application development tool for the Windows environment. You can create forms to query and update tables, insert new

records, and create tables in an Oracle database. It will only connect to Oracle databases, however, and there is no charting capability. Therefore, the demonstration applications were not developed for this product.

6 Conclusion

The difficulty in defining what is or isn't an Executive Information System extends to selecting one software as the "best" development environment. What may be a major concern for one developer could well be insignificant for another. It is possible, nevertheless, to make some general recommendations. Of this group, Microsoft Access performed better in more categories than any other package. It is also capable of development activities far beyond EIS applications. In the critical "look-and-feel" domain, it was a close second to Forests And Trees. Although it is somewhat more demanding of the developer than Forests And Trees, its other advantages make it a better choice for most situations.

Forests And Trees was the leader in visual appeal. If the intended user audience is small and located in one place, and the need is for a presentation-only system, Forests And Trees may be the best choice. Its automatic generation of drill-down and information-tree functions make development a fast and easy process.

The choice is less clear beyond these two. Lightship has some strong features, but it requires more programming resources than any other tool in this evaluation, and distribution to large numbers of users could be quite expensive. Of the spreadsheets, Excel is more capable, but if you already have Quattro Pro, you could develop a creditable system using it. Excel and Quattro are both constrained by the spreadsheet metaphor, and the visual results are noticeably less than with Access, Forests And Trees, or Lightship, but most organizations already have spreadsheet software, so the cost would be minimized. Only Oracle Card was an obvious failure for EIS development.

Bibliography

Bird, Jill. (1991). *Executive information systems management handbook.* NCC Blackwell Limited, Oxford, England, UK.

Burkan, Wayne C. (1991). *Executive information systems from proposal through implementation.* Van Nostrand Reinhold, New York, NY.

Moynihan, Gary P. (1993). "An executive information system: Planning for post-implementation at NASA," *Journal of Systems Management,* July 1993.

Rockart, John F. and De Long, David W. (1988). *Executive support systems.* Business One Irwin, Homewood, IL.

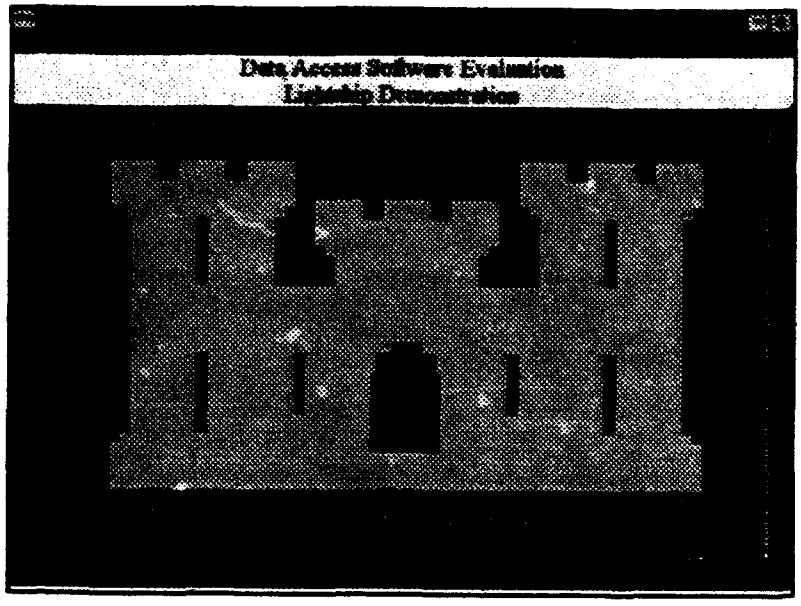
Thierauf, Robert J. (1991). *Executive information systems.* Quorum Books, Westport, CT.

Executive information systems. (1992). Hugh J. Watson, R. Kelly Rainer, and George Houdeshel, ed., John Wiley and Sons, Inc., New York, NY.

Appendix A
Tabulated Results

	Light-ship	Forest And Trees	Excel	Quattro Pro	Access	Oracle Card
Database Compatibilities	4	5	3	2	5	—
Database Access	3	5	3	2	5	—
Screen Development	3	4	2	2	4	—
Charting	4	4	4	3	4	—
Ease of Development	2	5	2	2	4	—
Ease of Maintenance	4	3	2	2	3	—
Distribution	2	2	3	3	5	—
Total Score	22	28	19	16	30	—

Appendix B
Application Screens



• Figure 1. Lightship main menu screen

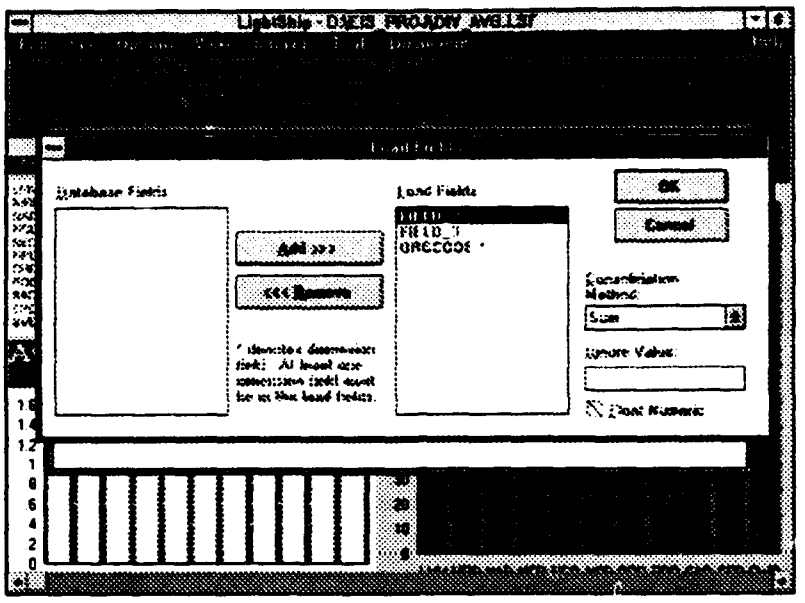


Figure 2. Lightship lens load fields screen

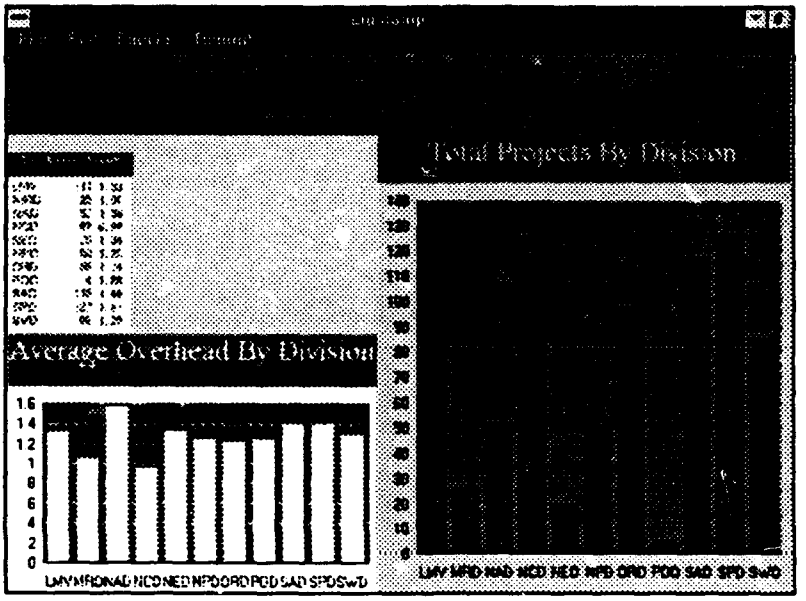


Figure 3. Lightship division overhead screen

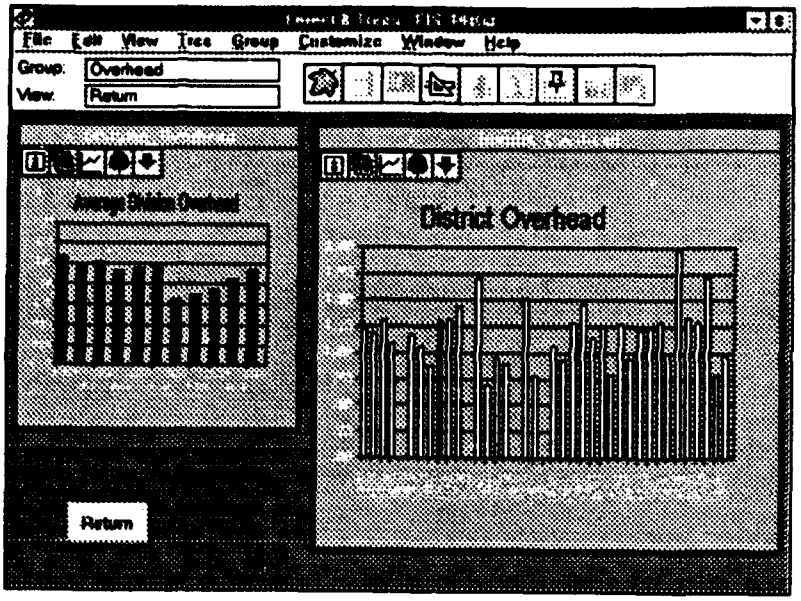


Figure 4. Lightship district project costs

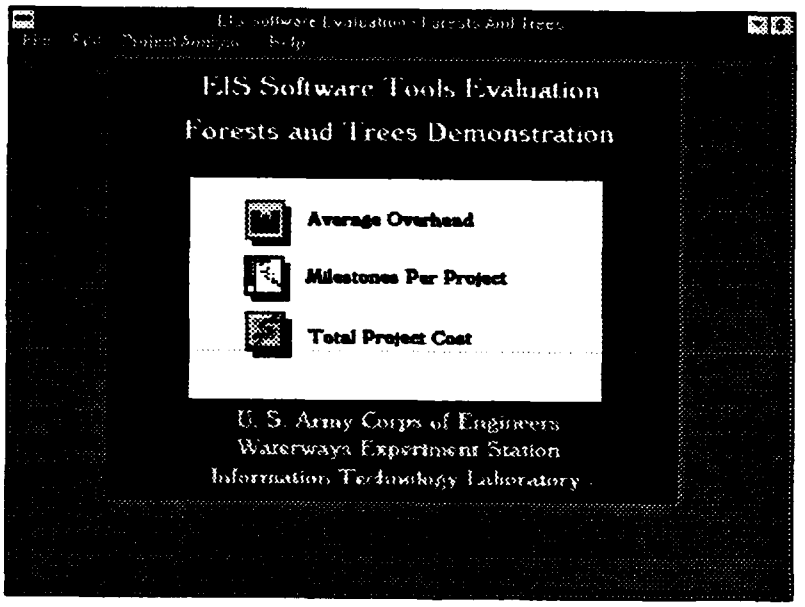


Figure 5. Forests And Trees main menu screen

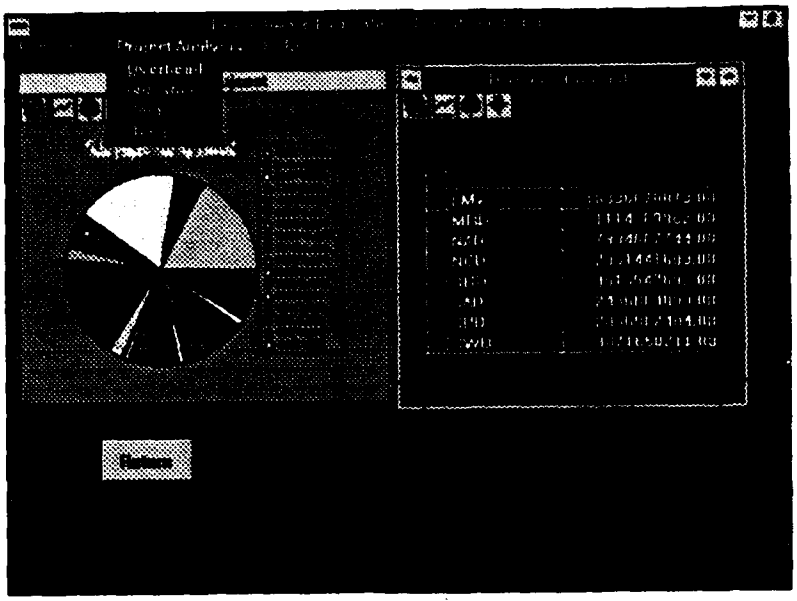


Figure 6. Forests And Trees project costs screen

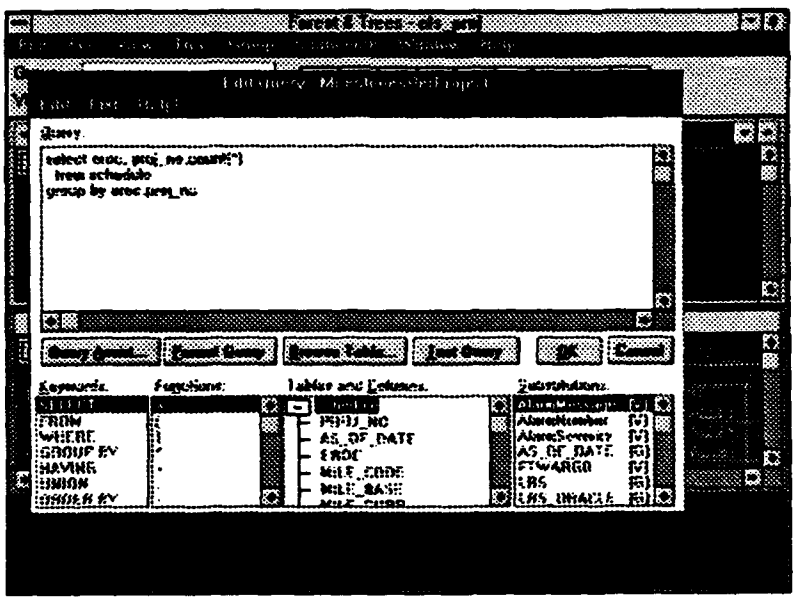


Figure 7. Forests And Trees query assist screen

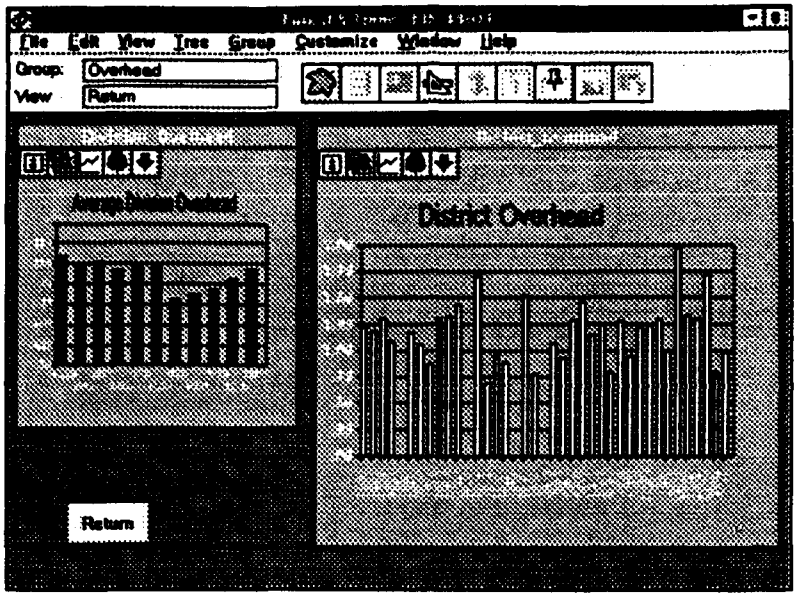


Figure 8. Forests And Trees average district overhead

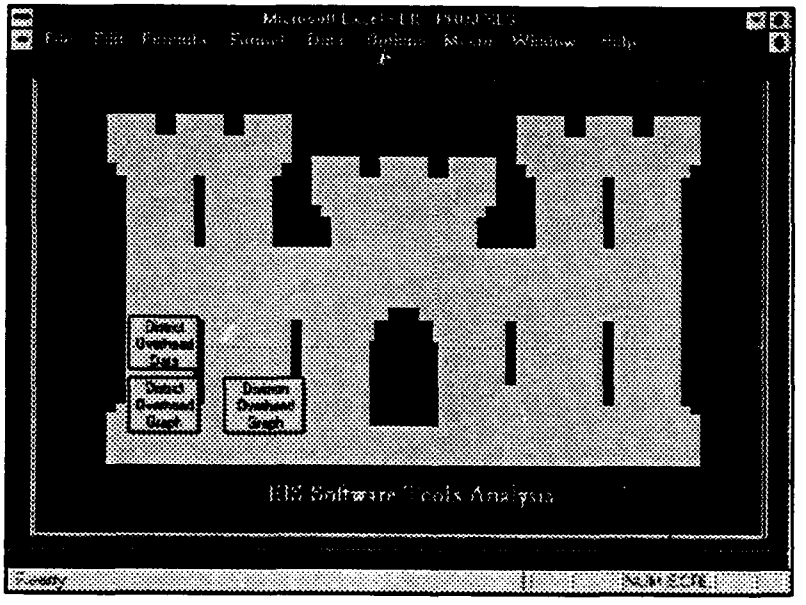


Figure 9. Excel main menu screen

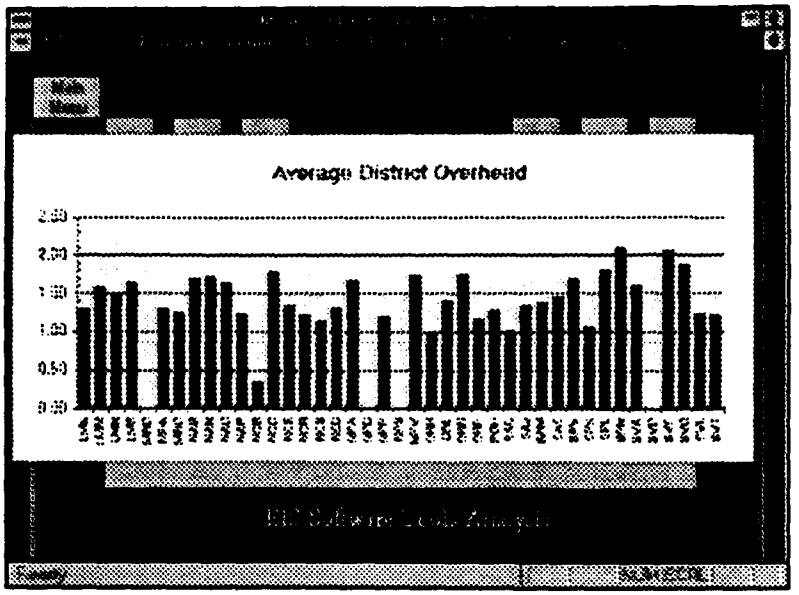


Figure 10. Excel district overhead screen

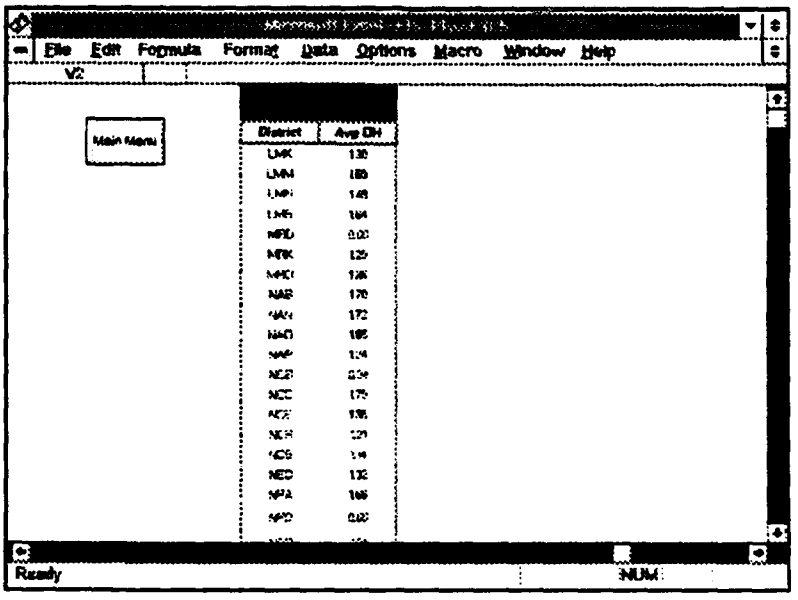


Figure 11. Excel district data

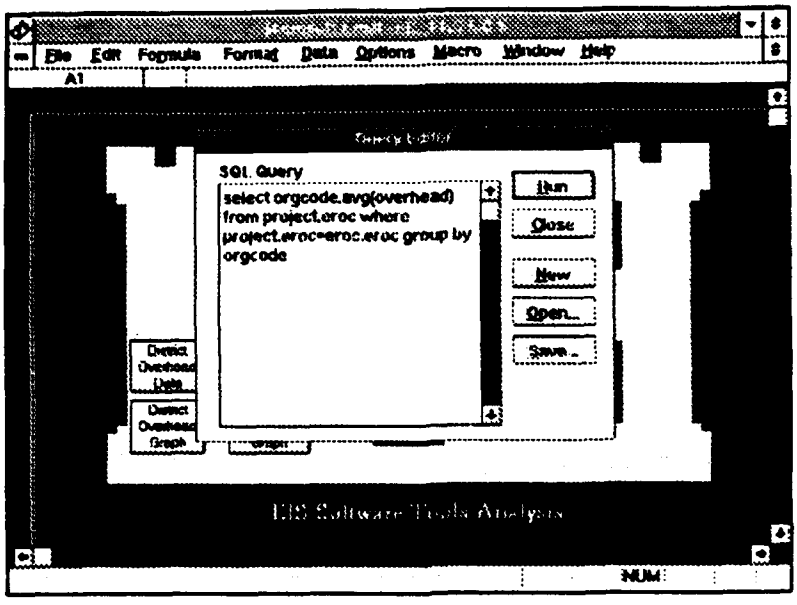


Figure 12. Excel Q+E interface

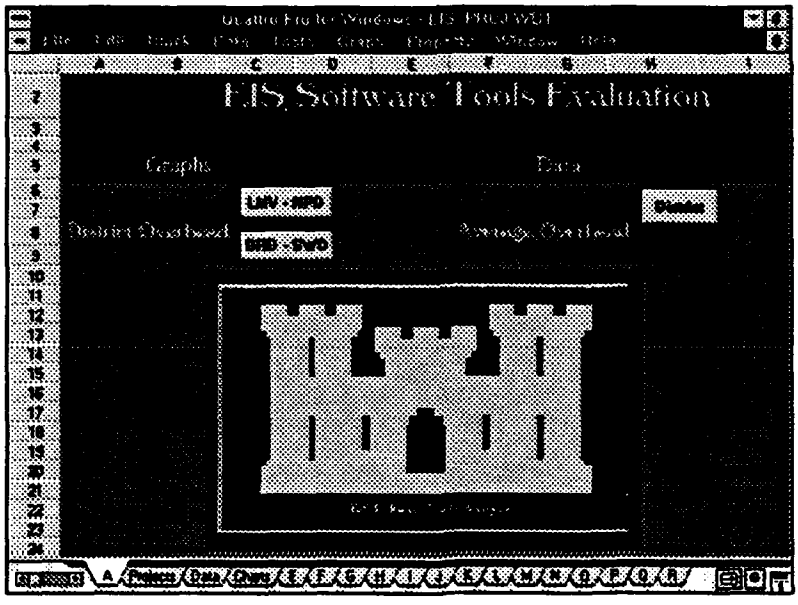


Figure 13. Quattro Pro main menu screen

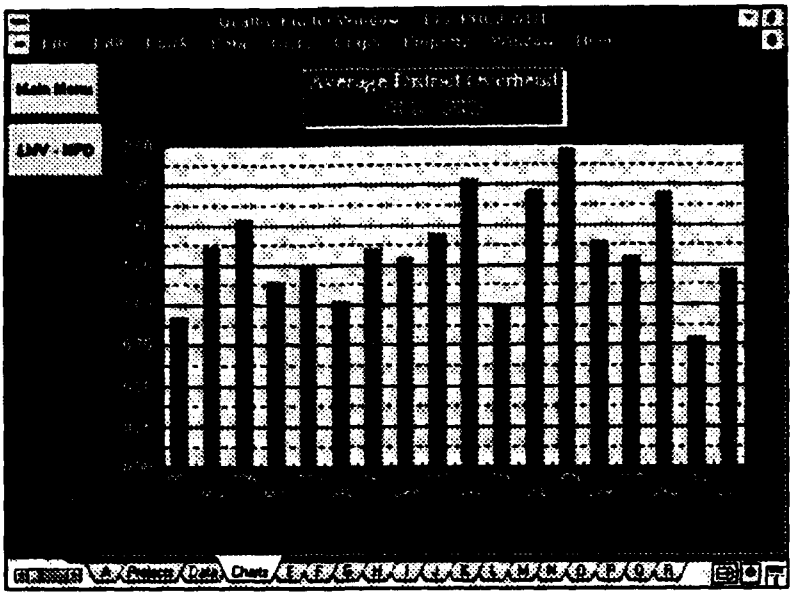


Figure 14. Quattro Pro district overhead screen

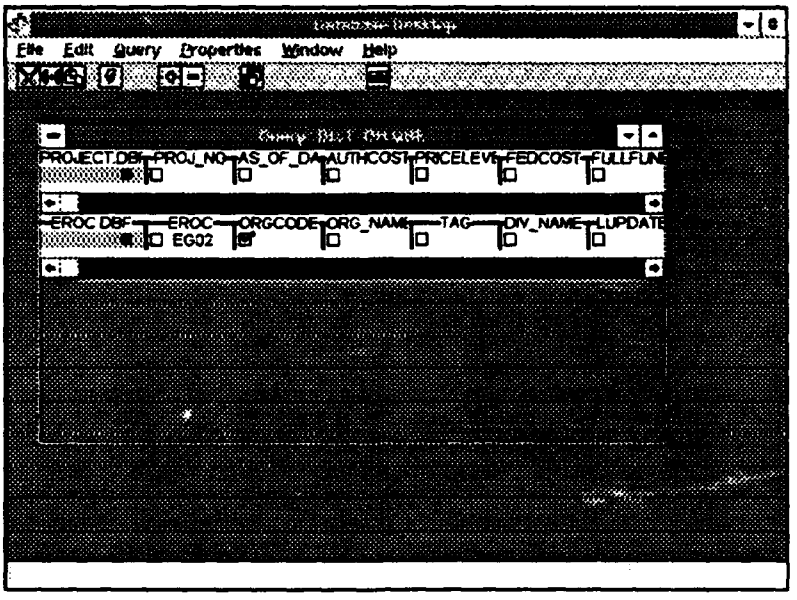


Figure 15. Quattro Pro database desktop

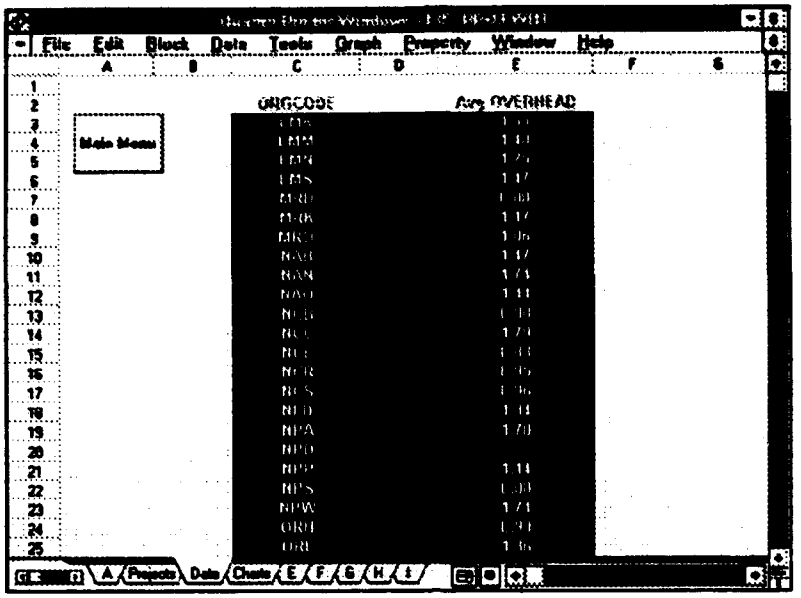


Figure 16. Quattro Pro district average overhead

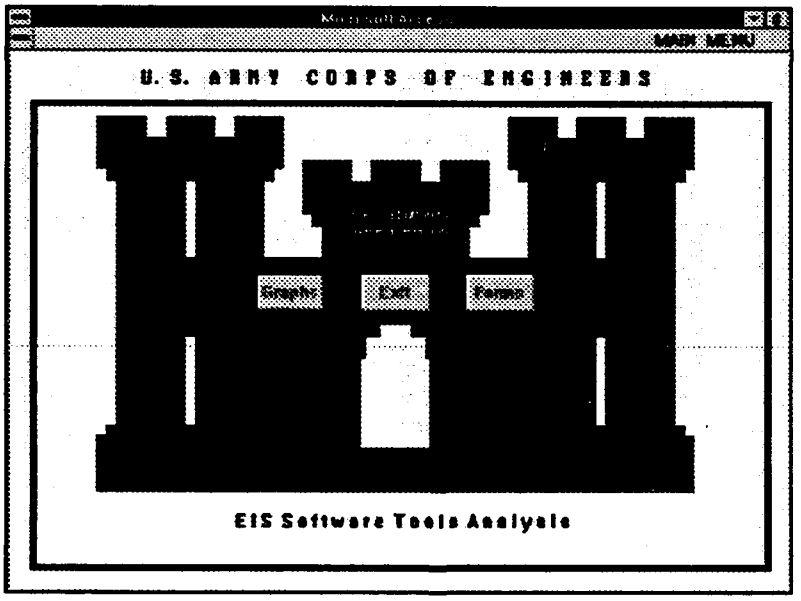


Figure 17. Access main menu screen

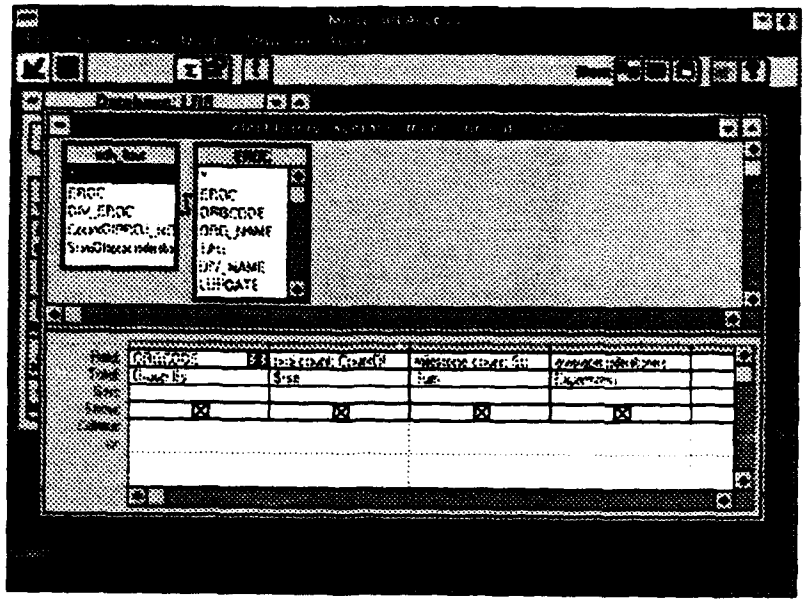


Figure 18. Access QBE screen

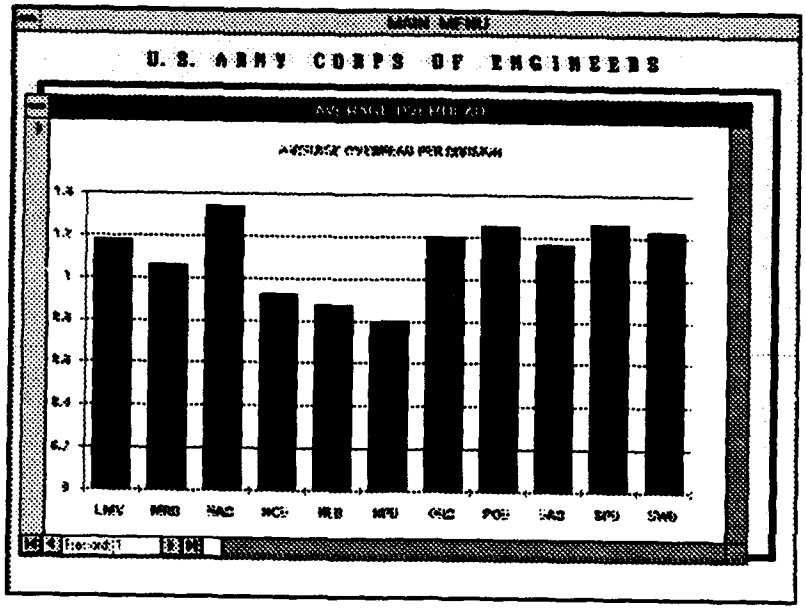


Figure 19. Access division overhead chart

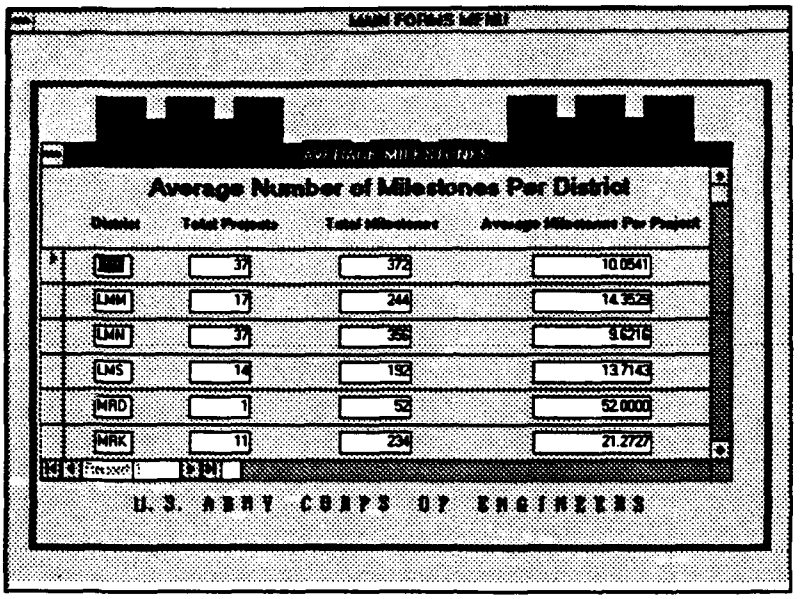


Figure 20. Access district milestones data screen

Appendix C
Database Compatibilities

	Light-ship	Forests And Trees	Excel	Quattro Pro	Access	Oracle Card
dBASE	Y	Y	Y	Y	Y	N
Paradox	Y	Y	Y	Y	Y	N
Lotus 123	Y	Y	Y	Y	Y	N
Excel	Y	Y	Y	Y	Y	N
Oracle	Y	Y	Y	N	Y	Y
SQL Server	Y	Y	Y	N	Y	N
Netware SQL	Y	Y	N	N	N	N
Btrieve	N	Y	N	N	Y	N
RDB	N	N	Y	N	N	N
R:BASE	N	Y	N	N	N	N
DataEase	N	Y	N	N	N	N
FoxPro	N	N	N	N	Y	N
DB2	Y	N	N	N	N	N
DDE	Y	Y	Y	Y	Y	N
ASCII	Y	Y	Y	Y	Y	N
Other	1	3	N	N	2	N

Appendix D
Demonstration Application Data Dictionary

Structure for table: d:\eis_proj\project.dbf
Number of data records: 778
Date of last update: 07/26/93
Memo file block size: 512

Field	Field Name	Type	Width	Dec	Index
1	PROJ_NO	Character	6		
2	AS_OF_DATE	Date	8		
3	AUTHCOST	Numeric	10		
4	PRICELEVEL	Date	8		
5	FEDCOST	Numeric	10		
6	FULLFUND	Numeric	10		
7	PROJ_NAME	Character	60		
8	EROC	Character	2		
9	MANAGER	Character	30		
10	OFF_SYMB	Character	15		
11	PHONE	Character	12		
12	EST_TYPE	Character	1		
13	EST_DATE	Date	8		
14	APPR_DATE	Date	8		
15	CURR_DATE	Date	8		
16	FORE_DATE	Date	8		
17	SCHED_DATE	Date	8		
18	SYNOPSIS	Memo	10		
19	CAT_CODE	Character	3		
20	UPLOAD	Logical	1		
21	APPN_CD	Character	2		
22	NFEDCOST	Numeric	10		
23	TOTPROJCST	Numeric	10		
24	OVERHEAD	Numeric	6	2	
25	NETMULP	Numeric	6	2	
26	DIV_EROC	Character	2		
27	ALTUSERIDS	Memo	10		
28	PROJ_TYPE	Character	2		
29	PROJ_PHASE	Character	1		
30	CONG_ADD	Logical	1		
31	HQ_ACTION	Logical	1		
32	CHANGED	Logical	1		
33	LUPDATE	Date	8		
34	LUPDATE_ID	Character	15		
** Total **			302		

Structure for table: d:\eis_proj\overbudg.dbf
Number of data records: 680
Date of last update: 07/26/93

Field	Field Name	Type	Width	Dec	Index
1	PROJ_NO	Character	6		
2	AS_OF_DATE	Date	8		
3	EROC	Character	2		
4	APPN_CD	Character	2		
5	PED_BASE	Numeric	8		
6	FED_BASE	Numeric	8		
7	NFED_BASE	Numeric	8		
8	PED_APPR	Numeric	8		
9	FED_APPR	Numeric	8		
10	NFED_APPR	Numeric	8		
11	PED_CURR	Numeric	8		
12	FED_CURR	Numeric	8		
13	NFED_CURR	Numeric	8		
14	PED_FORE	Numeric	8		
15	FED_FORE	Numeric	8		
16	NFED_FORE	Numeric	8		
17	PED_MVAR	Numeric	8		
18	FED_MVAR	Numeric	8		
19	NFED_MVAR	Numeric	8		
20	PED_TVAR	Numeric	8		
21	FED_TVAR	Numeric	8		
22	NFED_TVAR	Numeric	8		
23	LUPDATE	Date	8		
24	LUPDATE_ID	Character	15		
** Total **			186		

Structure for table: d:\eis_proj\budget.dbf
Number of data records: 4366
Date of last update: 07/26/93

Field	Field Name	Type	Width	Dec	Index
1	PROJ_NO	Character	6		
2	AS_OF_DATE	Date	8		
3	EROC	Character	2		
4	ACCT_CODE	Character	4		
5	PED_BASE	Numeric	8		
6	FED_BASE	Numeric	8		
7	TOTAL_BASE	Numeric	8		
8	PED_APPR	Numeric	8		
9	FED_APPR	Numeric	8		
10	TOTAL_APPR	Numeric	8		
11	PED_CURR	Numeric	8		
12	FED_CURR	Numeric	8		
13	TOTAL_CURR	Numeric	8		
14	PED_FORE	Numeric	8		
15	FED_FORE	Numeric	8		
16	TOTAL_FORE	Numeric	8		
17	APPN_CD	Character	2		
18	ORIG_EROC	Character	2		
19	LUPDATE	Date	8		
20	LUPDATE_ID	Character	15		
** Total **			144		

Structure for table: d:\eis_proj\schedule.dbf
Number of data records: 18520
Date of last update: 07/26/93

Field	Field Name	Type	Width	Dec	Index
1	PROJ_NO	Character	6		
2	AS_OF_DATE	Date	8		
3	EROC	Character	2		
4	MILE_CODE	Character	4		
5	MILE_BASE	Date	8		
6	MILE_CURR	Date	8		
7	MILE_FORE	Date	8		
8	MILE_ACT	Date	8		
9	APPN_CD	Character	2		
10	TWOA_CODE	Character	4		
11	ORIG_EROC	Character	2		
12	LUPDATE	Date	8		
13	LUPDATE_ID	Character	15		
** Total **			84		

Structure for table: d:\eis_proj\status.dbf
Number of data records: 605
Date of last update: 07/26/93
Memo file block size: 512

Field	Field Name	Type	Width	Dec	Index
1	PROJ_NO	Character	6		
2	AS_OF_DATE	Date	8		
3	EROC	Character	2		
4	FEASCH	Numeric	3		
5	FEAACT	Numeric	3		
6	DESSCH	Numeric	3		
7	DESACT	Numeric	3		
8	CONSCH	Numeric	3		
9	CONACT	Numeric	3		
10	APPN_CD	Character	2		
11	TESTI_CONG	Numeric	10		
12	TESTI_COMP	Date	8		
13	TESTI_FY	Numeric	2		
14	NOTEPAD	Memo	10		
15	L_CONG_PED	Numeric	10		
16	LC_PED_COM	Date	8		
17	LC_PED_FY	Numeric	2		
18	FEASI_COMP	Date	8		
19	FEAS_CMPFY	Numeric	2		
20	TOTSCH	Numeric	3		
21	TOTACT	Numeric	3		
22	EXPSCH	Numeric	8		
23	EXPACT	Numeric	8		
24	CONGDIST1	Character	60		
25	SPONSOR	Character	60		
26	LUPDATE	Date	8		
27	LUPDATE_ID	Character	15		
** Total **			262		

Structure for table: d:\eis_proj\ac_codes.dbf
Number of data records: 91
Date of last update: 07/26/93

Field	Field Name	Type	Width	Dec	Index
1	EROC	Character	2		
2	ACCT_CODE	Character	4		
3	ACCT_DESC	Character	40		
4	RPT_LEV	Character	2		
5	LUPDATE	Date	8		
6	LUPDATE_ID	Character	15		
** Total **			72		

Structure for table: d:\eis_proj\mileston.dbf
Number of data records: 1868
Date of last update: 07/26/93

Field	Field Name	Type	Width	Dec	Index
1	EROC	Character	2		
2	MILE_CODE	Character	4		
3	MILE_DESC	Character	40		
4	RPT_LEV	Character	2		
5	LUPDATE	Date	8		
6	LUPDATE_ID	Character	15		
** Total **			72		

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1994	Final report

4. TITLE AND SUBTITLE
Development of Executive Information Systems

5. FUNDING NUMBERS

6. AUTHOR(S)
Ernest H. Boswell, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER
Technical Report
ITL-94-6

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Executive Information Systems have been one of the key information management objectives for many organizations for over a decade. As personal computer hardware and software capabilities have grown, the potential for a useful and meaningful system has increased dramatically. Specifically, as the Microsoft Windows environment has become a de facto standard for personal computer development environments, many different software tools and methods have become available. This report investigates a representative set of Microsoft Windows-based software products to determine their applicability to Executive Information Systems development projects. A sample application is developed using each product, and the results are tabulated.

14. SUBJECT TERMS
See reverse.

15. NUMBER OF PAGES
40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED		

14. (Concluded).

Client-server systems
EIS
Executive information systems
Relational database management systems
Software development